



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

Visual Encodings for Networks with Multiple Edge Types

Citation for published version:

Vogogias, A, Archambault, D, Bach, B & Kennedy, J 2020, Visual Encodings for Networks with Multiple Edge Types. in *Proceedings of the International Conference on Advanced Visual Interfaces (AVI '20)*, 37, ACM, pp. 1-9, International Conference on Advanced Visual Interfaces 2020, Ischia, Italy, 28/09/20. <https://doi.org/10.1145/3399715.3399827>

Digital Object Identifier (DOI):

[10.1145/3399715.3399827](https://doi.org/10.1145/3399715.3399827)

Link:

[Link to publication record in Edinburgh Research Explorer](#)

Document Version:

Peer reviewed version

Published In:

Proceedings of the International Conference on Advanced Visual Interfaces (AVI '20)

General rights

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.



Visual Encodings for Networks with Multiple Edge Types

Athanasios Vogogias
Edinburgh Napier University
t.vogogias@gmail.com

Benjamin Bach
University of Edinburgh
bbach@inf.ed.ac.uk

Daniel Archambault
Swansea University
d.w.archambault@swansea.ac.uk

Jessie Kennedy
Edinburgh Napier University
j.kennedy@napier.ac.uk

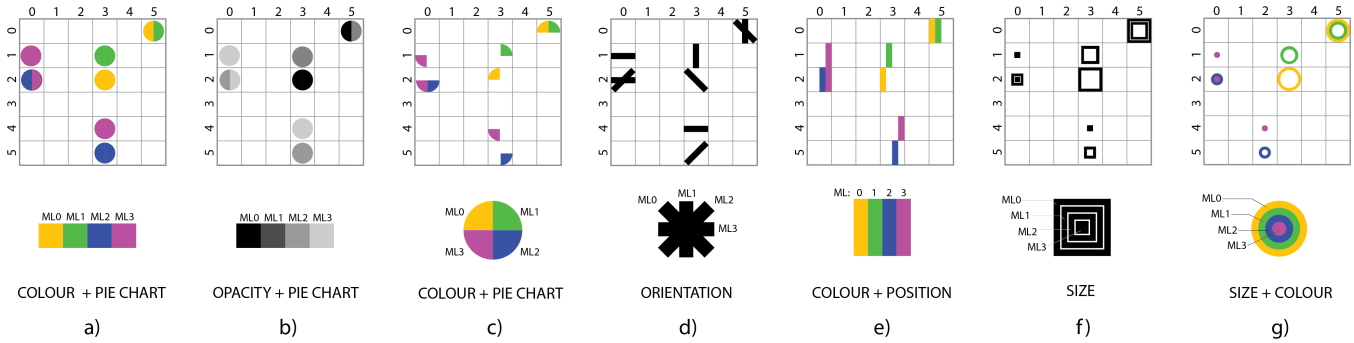


Figure 1: Examples of visual designs considered for encoding multiple types of edges in matrices. The top row shows an example of a single matrix and the bottom row shows the encoding for each edge type. The encodings use one or more visual variables to represent multiple edges: a) uses a coloured pie chart, b) uses opacity in a pie chart, c) uses a segmented and coloured pie chart d) uses orientation, e) combines position and colour, f) uses size and g) combines size and colour to create a glyph. The designs (d) and (e) were used in our experiments.

ABSTRACT

This paper reports on a formal user study on visual encodings of networks with multiple edge types in adjacency matrices. Our tasks and conditions were inspired by real problems in computational biology. We focus on encodings in adjacency matrices, selecting four designs from a potentially huge design space of visual encodings. We then settle on three visual variables to evaluate in a crowdsourcing study with 159 participants: orientation, position and colour. The best encodings were integrated into a visual analytics tool for inferring dynamic Bayesian networks and evaluated by computational biologists for additional evidence. We found that the encodings performed differently depending on the task, however, colour was found to help in all tasks except when trying to find the edge with the largest number of edge types. Orientation generally outperformed position in all of our tasks.

CCS CONCEPTS

• **Human-centered computing** → Visualization design and evaluation methods; User studies.

KEYWORDS

multilayer networks, adjacency matrices, dynamic Bayesian networks, evaluation, user study, crowdsourcing

ACM Reference Format:

Athanasios Vogogias, Daniel Archambault, Benjamin Bach, and Jessie Kennedy. 2020. Visual Encodings for Networks with Multiple Edge Types. In *AVI '20: International Conference on Advanced Visual Interfaces, September 28–October 02, 2020, Island of Ischia, Italy*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/1122445.1122456>

1 INTRODUCTION

Numerous visualisation techniques have been created and evaluated over the past decades to render complicated networks in a more understandable way. Surveys have been written about large networks [40], multilayer networks [9, 30], and dynamic networks [8], referencing techniques for dense networks with matrices, comparing graphs, tracking networks over time, as well as the techniques to more effectively communicate information and change in networks. Still, networks provide a large source of complexity and unsolved visualisation research questions.

In this paper, we explore visual encodings for multiple edge types in networks in the context of *dynamic Bayesian networks*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

AVI '20, September 28–October 02, 2020, Island of Ischia, Italy

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00

<https://doi.org/10.1145/1122445.1122456>

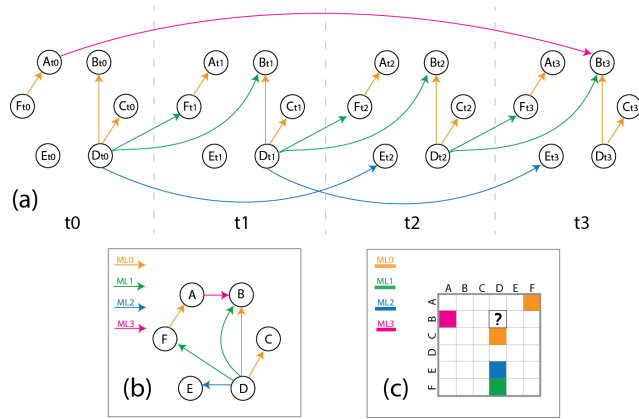


Figure 2: (a) The traditional representation of a DBN. The edge types of the DBN encoded without node repetition (b) as a node-link diagram and (c) as an adjacency matrix.

(DBNs) used in computational biology. Our collaborators in computational biology use DBNs to model probabilistic dependencies in complex biological processes. These processes range from gene regulation [45] to brain connectivity [36] and ecological networks [1]. Visualisation and comparison of these networks are crucial, e.g., for comparing edge types and alternative networks. For various reasons discussed with our collaborators, including familiarity in computational biology and strong potential for the efficient visual encoding of cells (squares instead of lines) [2], visualising DBNs calls for the use of adjacency matrices [39]. However, the design space for showing multiple edges of different types in matrices is potentially very large, including simple visual variables such as contrast [5], simple designs such as barcharts [19], subdivisions [7], or more carefully designed glyphs [12, 41].

In this paper, we contribute *i)* a structured exploration of cell-glyph designs for matrices (Figure 1), *ii)* present the results of a crowdsourcing study on those representations, and *iii)* report feedback from a lab evaluation with experts' analysis of DBNs. Our glyph designs include visual variables such as position, orientation, and colour, resulting in 9 designs, of which we tested the 4 most promising ones in our study. Our results show that encodings performed differently depending on the type of task. Colour was beneficial for all tasks that required users to distinguish between edge types, except the task where users had to identify cells with many types of edges without regard to their type. The final two designs were implemented in BayesPiles [39], a tool to provide visualisation support for Bayesian network inference, and evaluated by computational biology experts. We conclude with design recommendations based on our results.

2 BACKGROUND AND RELATED WORK

This section introduces dynamic Bayesian networks (DBNs), their representation as networks with multiple edges and reviews existing techniques to visualise networks with multiple edge types.

2.1 Dynamic Bayesian Networks

A Dynamic Bayesian Network (DBN) is a Bayesian network which relates variables to each other over adjacent time slices. Node-link representations are often used to visualise them [31] where all transitions are repeated in every time slice. An example DBN is shown in Figure 2(a) which we use to support our definitions. In this representation, time flows left to right. An edge connects nodes across different time slices, indicating the latency of an interaction called a *Markov lag* [43]. Markov lags are discrete and ordered based on their duration (*i.e.* the number of time slices skipped before an interaction is observed). For example, in Figure 2(a) the edge that connects F_{t_0} with A_{t_0} is of type 0, because it occurs in the same time slice, while the edge that connects A_{t_0} with B_{t_3} is of type 3 as the interaction extends 3 time slices. DBNs can become very long when they extend many time slices and can, therefore, be hard to visualise in this representation. More compact representations require another encoding to help distinguish between the different Markov lags. Figure 2(b) shows how the network becomes much smaller when nodes are not repeated and edge types encoded using colour, but these representations can suffer from visual clutter [2, 23]. In this paper, we look at visual encodings for multiple sets of edges on the same node set. Each set of edges is an *edge type* (*i.e.* each different Markov lag is an edge type).

Figure 2(c) shows how the same DBN can be represented as a directed adjacency matrix, again using colour to encode edge types. Edges are represented by adjacent cells in the matrix read by column and row (*i.e.* top-down). This representation is less cluttered than node-link diagrams when the networks are dense [2, 23], but when there are multiple edge types for the same edge, we need another encoding (Figure 2(c) illustrates the problem). A solution is a glyph that is able to represent the multiple edge types. In this paper, we explore the design space of such glyphs for these matrices and we evaluate their effectiveness for performing four analysis tasks.

2.2 Visualising Multilayer Networks

Several techniques have been proposed to represent and explore multilayer networks as node-link diagrams. Most of these methods employ colour and size as visual variables on nodes and edges [3, 4, 13, 33, 38, 42]. Other techniques include auxiliary visualisations such as bar charts or coordinated and multiple views combined with interaction [16, 34], or aggregate novel visual designs to convey link connectivity [18, 35]. Edge uncertainty has been explored for node-link diagrams which shares some aspects of this problem [24].

To overcome known problems such as visual clutter in node-link diagrams, matrices have also been used to visualise weighted edges [14] and to compare two weighted networks [2]. Matrices have further been used to visualise change over time for weighted edges [5, 6]. However, it is unclear how these representations can directly scale to support more than two edge types. Although the effectiveness of matrix visualisations has been shown empirically [22, 32], the potential of matrices for representing multilayer networks with different edge types has not been formally evaluated.

Glyphs provide a promising way to encode multivariate data in the cell of a matrix. Glyph design is well studied [10, 20, 41] and has been used for graph motif simplification [17] in node-link diagrams. In matrices, they have been used to encode attribute patterns on

edges [11] and in general information visualisations [12]. In this study, we only considered integrated approaches for matrix-based representations to extend the capabilities of an existing analytical tool called BayesPiles [39]. We explore glyph representations as they are scalable to several edge types and can be used with matrix representations.

3 FROM COMPUTATIONAL BIOLOGY TO ABSTRACT NETWORK TASKS

From four, one-hour-long interviews with our domain experts on their use of DBNs in understanding biological networks, we identified generic analysis tasks for networks with multiple edge types. We grouped these tasks into four levels, ranging from the simplest level (an individual edge in a single network) to comparing edges across multiple aggregated networks.

- **Edge tasks** describe tasks that look at identifying and describing individual edges and their edge types. For example, computational biologists are interested in the dynamics of a DBN based on its edges. When an interaction appears in multiple Markov lags (i.e., edge types), then there is uncertainty about the actual dynamics of that interaction in the system.
- **Network tasks** describe tasks that look at identifying edge patterns across a whole network. For example, time granularity is related (and often coincides) to the sampling rate picked for collecting expensive experimental measurements from the system. If most of the edges in a network are of Markov lag 3, time granularity is probably too fine; most edges being of Markov lag 0 implies the opposite.
- **Edge type co-occurrence** refers to tasks that require identifying co-occurrence of edge types within a network. For example, co-occurrence of edge types help to reason about the effect of the interactions of those Markov lags over time and determine if there are different biological processes captured in the same network that have different dynamics but affect the same pairs of nodes.
- **Network comparison tasks** include tasks that require comparing edges and Markov lags across networks. For example, to identify which network contains the highest number of a particular Markov lag or which one mostly contains Markov lags of a particular type. Computational biologists could then choose to include networks with uniform Markov lags as opposed to ones with a lot of variation (and therefore uncertainty).

The above analysis tasks show the importance for computational biologists to identify different edge types in visual representations of DBNs. Yet, any of these tasks can be applied to any network with multiple edges; these tasks are related with the ability to distinguish between edge types, identify instances and combinations of them, count them and compare them.

4 MATRIX CELL DESIGNS

The goal of our design was to find representations that can visualise multiple types per edge in the limited space of a matrix cell. Based on the task identification, perceptual principles, computational biologists' feedback and a literature review, we developed potential visual encodings for representing DBNs with matrices (Figure 1).

4.1 Design Criteria

- **C1: Number of edge types:** For matters of scalability, we expected that a successful encoding would be able to represent clearly at least *four* different edge types in a single cell, as identified important in our interviews with the experts.
- **C2: Design Simplicity:** given the limited amount of space in each matrix cell, we aimed for simplicity in design, tried to maximise the *data-to-ink ratio* [37], and base the design of those encodings on perceptual principles [28] about primary visual variables, such as colour, opacity, size, shape, texture, orientation, position, and their combinations to create glyphs [21, 29, 41].
- **C3: Edge type ordering:** As Markov lags are numbered (0-3), we only considered designs in which edge types could be perceived ordered and that can work both with and without colour. Although ordering is inherent in the different Markov lags, identification of particular Markov lags was more important in the tasks. Therefore, we used categorical colouring as double encoding to test its effect on encodings and make our study more generalisable, as colour could be used to map a different attribute in other domains.

4.2 Design Space

Figure 1 shows seven of our nine initial designs. The upper row in this figure shows the encodings for a single matrix where we need to visualise multiple edge types. For example, design (a) uses colour and design (b) uses shades of grey to differentiate types of edges. Design (c) is a variation of design (a), using equally-spaced and coloured segments of a pie chart to indicate the presence of a lag (one segment per lag type). Design (d) uses orientation (angle) to differentiate edge types and encode their order. Design (e) uses position and colour within each cell, resulting in a striped cell design. Design (f) uses size and design (g) is a variation of design (f) which uses both size and colour to encode category and order, but instead of superimposed squares [2], it uses rings.

4.3 Discussing Designs

Through discussion, we rejected designs that would easily become cluttered when multiple edge types were combined in the same cell and those that did not make it easy to distinguish between the different types of edges. For example, we rejected designs (a) and (b) because edge types that coexist in the same cell are represented in a smaller area than in cells that contain only one edge type. Design (c) (which was used by Dang et al. [15]) was found more intuitive than the segmentation of the cell into rectangles [27], but pie segments without colour become illegible within the limited cell space. On the other hand, using orientation or position as a primary variable, as in designs d) and e), results in more discernible edge types even when colour is not used. Alper et al. [2] found that colour was not necessary for f) and g) and that the effectiveness of the techniques is all about contrast. As of colour-area bias, we rejected designs f) and g) with colour. We also rejected designs in which it was more easy to identify one edge type compared to another, such as designs (f) and (g) without colour, because for our tasks the visual variable of size introduced visual bias that favoured edge types of larger sizes (easier to spot) compared to edge types of smaller sizes. Apart from the 7 encodings shown in Figure 3, we also considered a variation

of (d) that combined orientation with colour and a variation of (e) that used position without colour (9 designs in total). We decided to test designs that can work both with and without colour as a double encoding and where visual marks for each edge types have the same size. These designs were d) and e). This resulted in the *four* most promising encodings to compare in our final study (Figure 3).

4.4 Final Designs

Examples of the final encodings used on the same multilayer network are shown in Figure 4. The selected encodings use at least one of the following visual variables: *position*, *orientation* and *colour hue*. Orientation and position can stand alone or in combination with colour, however, colour can only be used in combination with another visual variable. We used ColorBrewer [26] to provide colours that looked balanced and distinct.

Theoretically, when a five-degree change in the angle is used, the approach could scale to encode up to 36 edge types. However, it is unrealistic to expect that users can effectively distinguish between a forty-degree and a forty-five-degree angle. Similarly, position can also encode more edge types by slicing the cell in thinner stripes. Colour can scale to encode up to 8 discernible edge types of similar intensity [26].

All four selected encodings could *sufficiently* represent networks with at least four multiple edge types, such as found in our collaborators DBNs. However, we wanted to compare the encodings and find which one was the most *effective* for supporting the identified analysis tasks.

For each task we created the following two hypotheses:

- **H1:** Encodings using *orientation* outperform encodings using *position*. We believe that orientation is more salient.
- **H2:** Redundantly encoding edge type with colour will improve the performance of either encoding (*position* or *orientation*).

To test our hypotheses, we ran a quantitative evaluation study to assess the effectiveness of the final encodings. The study involved participants from the general public who were asked to perform simple visual analysis tasks on matrices that use those encodings.

5 USER STUDY

To test the effectiveness of the encodings in multilayer networks represented as matrices, we performed four controlled user studies through Amazon Mechanical Turk (AMT). The visualisation tasks that participants were asked to complete in the study were derived from the tasks described in Section 3. Each experiment evaluated the encodings with a different task, with each participant performing only one task at a time. Participants could perform each task (experiment) only once, but if they wanted they could participate in all four of them. We present the results of the four experiments individually and then discuss the overall results.

5.1 Data Generation and Pilot Studies

In discussion with our computational biology collaborators, it was clear that their real-world data sets were limited to four edge types in networks of around 30-50 nodes, which are also common testing conditions in network evaluations [44]. To keep study complexity manageable while assuring the validity of results, our studies

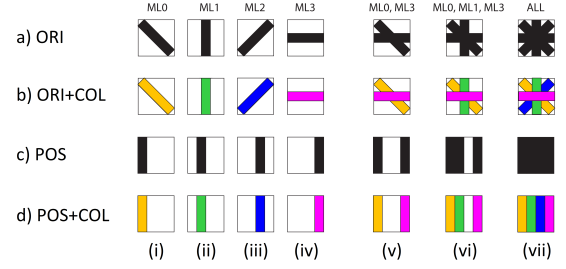


Figure 3: Proposed visual encodings for the different types of edges tested in the user study: a) orientation without colour (ORI), b) orientation with colour (ORI+COL), c) position without colour (POS) and d) position with colour (POS+COL). Columns (i), (ii), (iii) and (iv) show the encoding of Markov lag 0, 1, 2 and 3 respectively. Columns (v), (vi) and (vii) show how the combination of two, three and four edge types look in the different encodings.

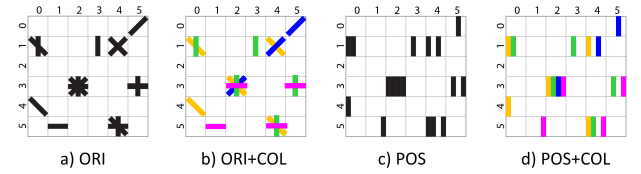


Figure 4: An example of the final encodings showing the same multilayer network: a) orientation without colour (ORI), b) orientation with colour (ORI+COL), c) position without colour (POS) and d) position with colour (POS+COL).

tested only four edge types. We therefore generated Watts-Strogatz networks of two sizes (30 and 50 nodes) using NetworkX [25], to target a wide range of real-world networks which often exhibit small-world properties. The probability of a random edge (p) was set to 0.25 and to avoid disconnected networks, every node was initially connected to its 4 neighbouring nodes ($k = 4$). These networks had similar properties to those found in our application domain of multilayer networks. To generate multiple trials we randomly removed 15% of edges resulting in networks of similar structure with the same number of connections for each trial. We also extended the algorithm to generate multilayer networks of four edge types. The distribution of percentages for every combination of edge types could be controlled in each network. Thus, the number of cells that contained a particular combination of edge types was always known. This network generation process ensured precision and control over the properties of the networks so that each data set would consist of different networks with similar characteristics, such as structure, size, regularity, connectivity and difficulty.

We performed a series of pilot studies to find data sets of realistic sizes. In our pilots, we tested networks of two sizes (of 30 and 50 nodes) and four difficulty levels (“very easy”, “easy”, “hard” and “very hard”). To increase the difficulty level, we incrementally added distractors to the data set, and the number of distractors added in each difficulty level varied depending on the task. For instance, in each trial of the first task, there was only one correct

answer (*i.e.* a cell that contained 3 or 4 edge types) and distractors were considered the cells that contained just one edge type less than the correct answer (*i.e.* 2 or 3 edge types respectively). Every participant in the pilot study performed one task in all 4 difficulty levels to identify the correct level of difficulty to avoid floor and ceiling effects. We found that participants had high response times and error rates for “hard” or “very hard”. In those difficulty levels, the number of distractors was more than 50% of the total edges in the entire network, a condition which is unusual in real data sets. Therefore, we only used “very easy” or “easy” networks of 30 nodes each, which corresponds to typical real-world data sets. Also, we observed that using the grid helped participants perform better in all tasks. Therefore, we chose to use the grid in all our experiments. In particular, participants reported that the grid helped them distinguish between glyphs in neighbouring cells, especially for larger matrices (of 50 nodes) and when position was used in the encoding.

5.2 Study Tasks

For each level of task identified in Section 3, we chose a specific representative task and performed a crowdsourcing user study.

T1 (T-EDGE)—Find the edge with the highest number of edge types. The task required participants to (a) identify the densest cell in the matrix and (b) compare it to other dense cells in the same matrix to verify their choice. The participant would click on the cell with the most edge types. In each trial, the correct answer was only one cell in the entire matrix, and either contained all four edge types or three out of four edge types. Therefore, the possibility of clicking on the correct cell just by chance was very low.

T2 (T-NETWORK)—Which edge type appears most frequently in the matrix? Participants had to: (a) distinguish between the four edge types and (b) compare their frequency of appearance in the matrix. The participant selected an answer via a multiple-choice question. In each trial, the correct answer had a double number of instances compared to any other type and 25% of those instances appeared alone and the rest in combination with a second edge type distributed in equal numbers.

T3 (T-EDGE CO-OCCURRENCE)—Which combination of two edge types appears most often in the matrix? Participants had to: (a) distinguish between the 6 possible combinations of two edge types and (b) compare their frequency of appearance in the matrix. In many ways, this task is similar to T2 but instead of looking for single edge types, we are interested in combinations of two edge types and their resulting visual encoding. The participant selected an answer via a multiple-choice question. In each trial, the correct answer had a double number of instances compared to any other combination of two edge types. Also, while there were cells with a single edge type or combinations of more than two edge types, they appeared in the same proportion.

T4 (T-NETWORK COMPARISON)—Which matrix has more edges with edge type X? X being replaced with different types 0-3 for each trial.

Participants had to: (a) distinguish between the 4 edge types to find the one requested and (b) compare the frequency of its appearance between two matrices. This task is similar to T2 but involves the comparison of two matrices. Participants clicked on the matrix to indicate their response. Answer possibilities were:

“Both the same” and “I don’t know”, but those options were never the correct answer in the test. In each trial, the wrong matrix had the same percentage of either edge type. However, the correct matrix had an increased percentage of the requested edge type, while the rest of the edge types shared an equal percentage of instances. Thus, both matrices maintained the same density (*i.e.* the total number of edges).

5.3 Experimental Procedure

Each of the four user studies tested one task in a within-subject 2×2 design ($\times 2$ techniques: orientation/position $\times 2$ colour schemes: with/without colour), testing the following 4 experimental conditions (*i.e.* encodings in Figure 3): a) orientation without colour, b) orientation with colour, c) position without colour and d) position with colour. For each task we recruited around 45 participants through AMT, resulting in a total of 177 participants. After removing invalid participants, we were left with 159 valid. Average age was 37, male-female ratio was 66% male and 34% female. About half of the participants did not have any experience with data visualisation; only about 7% used visualisations on a daily basis.

The first page of each study contained information about the study and a list of terms which participants were asked to read carefully. Then participants were asked demographic questions such as: age, gender, familiarity using data visualisations, and the device they were using for the study (laptop, desktop, tablet or phone). In the case they selected phone, the website showed a message that participation was not possible due to the small available screen size. Throughout each study, the entire information was shown in one full-screen window without a need to zoom or pan.

Each study had 4 *blocks* (parts) of trials, one for each encoding. Before each block, there was a page with instructions that explained the task and demonstrated how to use the interface to submit answers. Participants were asked to first complete the task on a demo trial before proceeding to the main study, to familiarise themselves with the interface. Following feedback explained why their response was correct or incorrect. Except for the demo trial at the beginning of each block, participants did not receive feedback on the correctness of their answers in the recorded trials.

After the demo trial, each block contained 3 training trials, followed by 12 recorded trials for tasks T1 and T3 and 8 recorded trials for tasks T2 and T4. Fewer trials were used for tasks that participants spent more time to complete in our pilot studies. Also for tasks T2, T3 and T4, an attention (gold standard) trial was placed randomly within the recorded trials showing a very simple situation where an attentive participant would not be expected to make an error. Attention questions were similar to the recording trials with the only difference that the correct answer was very easy to find. For example, for T2 the correct answer would appear in a very high percentage compared to all other answers (e.g. in 90% of the edges). Those trials were later used to identify participants who were not invested in the experiment or who did not understand the task. In task T1, there was no gold standard and the participant attention to the task was evaluated by comparing mean error rate with an error rate close to chance. In between trials, a neutral screen would appear for two seconds to help participants focus their attention on the next image.

The blocks of the experiment were counterbalanced to alleviate the effects of presentation order (learning and fatigue effects). Visual encodings were very different, precluding randomising experimental trials. For each of the four experiments, roughly half the participants started the experiment orientation first. Within each encoding (orientation and position), roughly half performed the task with colour first. Except for the order of blocks, the recorded trials were also shuffled within each block for each participant. Between the 4 blocks, participants were encouraged to take breaks.

The effectiveness of each encoding was evaluated through two dependent variables: (a) participants' *response time* (in milliseconds), that measured efficiency and their (b) *error rate*, that measured accuracy. For each trial, there was only one correct answer, so the average error rate of each block ranged between 0 and 1. However, regarding participants' response times, there was a danger of including outliers by taking the average, since we did not have control over the study environment. Therefore, for each encoding, the average response time was calculated as the 25% truncated (trimmed) mean of response times within each block of recorded trials. This would exclude the minimum and maximum response times of each block from calculating the mean. Thus, from the 8-12 trials in each block, we only considered 6-10.

Each study was run on Amazon Mechanical Turk (AMT), linking to our own JavaScript website. All participants were located in the US and paid \$3 for completing the test which lasted approximately 15 minutes. Participants were told to answer as fast and accurate as possible. There was no time limit to complete the study. Multiple participation in the same study was prohibited, but participants could participate in all four studies. At the end of the test, an authentication code would appear on screen which participants could use to claim payment through AMT for completing the task.

6 RESULTS

We analyse the results considering each task as a separate experiment. Our statistical protocol was set out in advance and applied separately to the response time and error rate of the four combinations of factors: ORI (Orientation without Colour), ORI+COL (Orientation with Colour), POS (Position without Colour), and POS+COL (Position with Colour). Before analysis, we applied data quality checks. We removed participants who (i) did not complete the full experiment, (ii) they had an average error rate close to chance (applied to T1), or (iii) they failed to answer more than half of the gold standard questions (applied to T2, T3 and T4).

After the quality checks, the response time and error rate data were analysed separately. For each of the four distributions, we used a Shapiro-Wilk test with a significance level of $\alpha = 0.05$ to determine if the data were normally distributed. We also plotted the distributions to visually check for normality. In all cases for response time and error, at least one of the distributions was found to not be normally distributed. Therefore, we ran a Friedman test with a significance level of $\alpha = 0.05$ to determine if there was a significant difference between the four distributions. We used a post-hoc approximative (Monte Carlo) Nemenyi-Damico-Wolfe-Dunn test to determine the pairwise significant differences. For all results bar charts (Figure 5), the margin of error for 95% confidence intervals is shown in each bar while the black lines between bars indicate

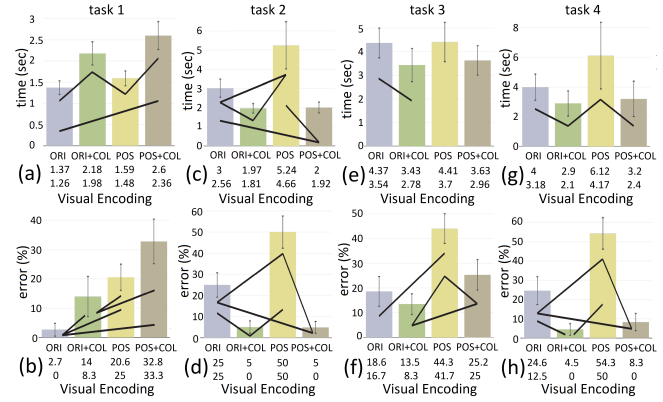


Figure 5: Mean (a) response times and (b) error rates for task 1. Mean (c) response times and (d) error rates for task 2. Mean (e) response times and (f) error rates for task 3. Mean (g) response times and (h) error rates for task 4. The lines indicate significance between visual encodings.

significance between visual encodings. The difference between bar heights indicates the magnitude (absolute effect size). Mean and median values are indicated below each bar.

T1 (T-EDGE)— Our first experiment asked: *Click on the cell with the most marks*. We collected the data from 42 participants in this experiment, but one was excluded as they did not complete the full experiment. We analysed the data of the remaining 41 participants. None of the participants gave answers with an average error rate close to chance (error rate of 98%).

Results. Figure 5 (a) and (b) shows our results. We find significant results for response time ($\chi^2 = 75.117$, $df = 3$, $p < 0.05$). Our post-hoc analysis reveals that ORI was significantly faster than both ORI+COL and POS+COL. POS was significantly faster than ORI+COL and POS+COL. Also, we find significant results for error rate ($\chi^2 = 71.032$, $df = 3$, $p < 0.05$). Our post-hoc analysis reveals that ORI produced significantly fewer errors than POS, ORI+COL, and POS+COL. ORI+COL produced significantly fewer errors than POS and POS+COL.

As POS was faster than ORI+COL but produced more errors, we performed a correlation analysis between response time and error. The analysis showed negative correlation between response time and error for both POS and ORI+COL distributions. This means that participants who answered faster tended to answer incorrectly. However, the error rate for POS was still low (near 20%).

Discussion. Our results are surprising as they provide evidence that colour significantly hurts the efficiency (response times) of participants completing the task in all cases (**reject H2**). For this task, users were not required to differentiate between types of marks across cells. Thus, colour seems to have added distracting complexity to the representation. Orientation consistently produces significantly fewer errors than position with no significant difference in response time (**accept H1**). As with most online studies, it is very hard to gather good qualitative data or observations of the participants through crowdsourcing. However, our expert evaluation, presented in Section 7, provides some further evidence for these interpretations.

T2 (T-NETWORK)— Our second experiment asked: *Which mark appears most often in the image?* In total, 50 participants performed experiment 2. From those 4 were excluded because of incomplete data and 1 because they used a phone. No participant had an average error rate close to chance (75%). We analysed the data of the remaining 45 participants.

Results. Figure 5 (c) and (d) shows our results. We find significant results for response time ($\chi^2 = 77.948$, $df = 3$, $p < 0.05$). Our post-hoc analysis reveals that ORI+COL is significantly faster than ORI and POS. Similarly, POS+COL is significantly faster than ORI and POS. We also found that ORI is significantly faster than POS. We find significant results for error rate ($\chi^2 = 96.746$, $df = 3$, $p < 0.05$). The pairs of results are exactly the same as those found in response time. Our post-hoc analysis reveals that ORI+COL is significantly more accurate than ORI and POS. Similarly, POS+COL is significantly more accurate than ORI and POS. We also found that ORI is significantly more accurate than POS.

Discussion. In general, the results show that for this task it is preferable to use colour for encoding the different types of edges (**accept H2**). For this task, it could be that colour provides a method for quickly gauging the number of marks of each type at a glance, allowing for the quick identification of a more prevalent one. When colour is not used, orientation seems to perform better than position (**accept H1**). It could be that since the marks for orientation have a more unique appearance, it is easier to judge the number of such marks in the matrix visualisation.

T3 (T-EDGE CO-OCCURRENCE)— Our third experiment asked: *Which combination of two marks appears more often in the image?* In total, 45 participants took part in experiment 3. When checking the data quality, 4 participants were excluded because of incomplete data and 1 did not answer at least half of the four gold standard questions. None of the remaining participants had an average error rate close to chance (84%). We analysed the data of the remaining 40 participants.

Results. Figure 5 (e) and (f) shows our results. We find significant results for response time ($\chi^2 = 14.91$, $df = 3$, $p < 0.05$). Our post-hoc analysis reveals only one significant difference with ORI+COL being significantly faster than ORI. In terms of error rate, we find significant differences ($\chi^2 = 45.608$, $df = 3$, $p < 0.05$). ORI+COL has significantly fewer errors when compared to POS+COL and POS. POS+COL has significantly fewer errors than POS. Also, ORI has significantly fewer errors than POS.

Discussion. The error rates indicate that orientation has significantly fewer errors than position in many cases (**accept H1**). When orientation is used, it either reduces the error rate with no difference in response times or it reduces response times with no difference in error rate (**accept H1**). Colour, when used in combination with orientation, seemed to reduce response times and when it was used in combination with position it reduced error rates (**accept H2**).

In this task, we asked users to select the pair of marks that occurred most frequently together. Colour may have helped the participant judge the frequency of each mark type at a global level and select the pair that occurred more frequently. As the orientation marks are more unique in appearance, it could have helped the participants make this judgement.

T4 (T-NETWORK COMPARISON)— Our fourth experiment asked: *Which matrix has the most cells of a particular mark type?* In total,

40 participants performed the experiment. From these participants, 3 failed more than half of the gold standard questions, 2 had incomplete data, and 2 had a variance of answers equal to zero – meaning they clicked on the same answer for the duration of the experiment. All these 7 participants were excluded. From the remaining 33 participants, no one had an average error rate close to chance (75%). We analysed the data from the remaining 33 participants.

Results. Figure 5 (g) and (h) shows our results. We find significant results for response time ($\chi^2 = 30.018$, $df = 3$, $p < 0.05$). Our post-hoc analysis reveals that ORI+COL was significantly faster than ORI and POS. We also find that POS+COL is significantly faster than POS. In terms of error rate, we find significant differences ($\chi^2 = 71.181$, $df = 3$, $p < 0.05$). ORI+COL has significantly fewer errors than ORI and POS. POS+COL has significantly fewer errors than ORI and POS. Finally, ORI has significantly fewer errors than POS.

Discussion. From the results, it is clear that colour provides the most benefit and helps participants perform this task (**accept H2**). When colour is not used, then orientation produces significantly lower error rates than position (**accept H1**). As the task requires the participant to make a global assessment of the presence of a particular mark in the matrix, colour could have provided a means to quickly gauge the number of marks. If no colour is present, the uniqueness of the marks in the orientation condition might have helped participants make this judgement.

7 EXPERT EVALUATION

To better understand our techniques with realistic data sets and tasks, we conducted an expert evaluation with three computational biologists (1 researcher, 2 PhD students). Based on the results of our experiment, we extended the design of BayesPiles to use orientation and enabled colour toggling (on/off).

7.1 Data Set Used in the Evaluation

To create a realistic collection of networks, we ran BANJO on the songbird data [36] to create 17 DBNs. This collection of networks was then modified to ensure known correct answers. For example, we inserted networks into the collection that were permutations of copies of the top-scoring network with single edge types and frequently occurring pairs of edge types added to keep density relatively constant. In one network, we inserted an edge with all four edge types. Also, networks in the solution space were edited to have three score levels with more edges of observed type combinations into those networks with the lowest score. The result was a curated collection of 30 realistic DBNs with patterns in the data set that had been observed and known correct answers, used to evaluate T1-T4 in a controlled way with our experts.

7.2 Evaluation

The single two-hour session including all three experts began with a brief demonstration of BayesPiles. Then, we distributed an evaluation form asking the experts to load the 30 network data sets into BayesPiles and asked them to perform the tasks with and without colour. Legends were included to specify the encodings of Markov lags. A specification of the task was provided, and participants were evaluated on the correctness of their answer. At the end of the form, participants could provide qualitative feedback and comments.

7.3 Results

Although the tasks the experts performed were the same as those tested in the crowdsourcing experiment, our experts were required to perform the tasks on piles of networks with weighted edges, making it more demanding; edge weight was encoded as lightness of the respective edge mark. **For T1** most participants visually inspected all 30 networks juxtaposed as small multiples and successfully found the densest cell in the whole collection. Participants that answered correctly preferred not to use colour for this task, confirming the results in our controlled study. *P2* commented that “it was easier to see a dense block of lines without colour. With colour, the ‘star’ of all blended into the vertical Markov lag 1 below it, and didn’t look so different from networks 4-8 which didn’t have the Markov lag 1 in the ‘star’ but did in the cell directly below it.”

For T2, all participants could identify which edge type was the most common in the matrix, but all participants found this task difficult. *P1* said, “it would be helpful if there was some information about the number of edges at different Markov lags, as it is easy to make mistakes if the edges have to be detected by eye, especially in larger networks”. For piles, the task was particularly demanding due to opacity. *P2* said “it was easier to compare different opacity levels in the grey scale, but the trade-off was that it was harder to distinguish between Markov lags when colour was not present.”

There was significant variation for **task T3**; finding the most common combination of Markov lags. Participants needed to estimate both weight in the pile and the frequency of the pairs and many pairs had low weights, appearing “faint” in the pile, making this one of the hardest tasks. This made it challenging for the participants to correctly estimate the overall weight of each combination of two Markov lags appearing in the matrix and then compare those estimations effectively.

Regarding T4, two experts could identify the pile that contained the highest concentration of a particular edge type. However, this task became much harder when experts were required to estimate the overall weight of an edge type in a pile of heterogeneous networks. Combining opacity with colour tends to affect the distinctiveness of the colour, which often becomes difficult to recognise for edges that have a lower weight. When colour is not used, it becomes even more challenging to both identify all instances of a particular type and also estimate their overall summary. *P2* commented that “I found this the hardest task. [...] It was easier to identify Markov lag 2 with colour, but then I was unsure if perhaps the shading might make there be less overall Markov lag 2, even though there was clearly more *kinds* of Markov lag 2”.

8 DISCUSSION

Effectiveness of Encodings—Considering all four experiments, ORI outperformed POS and colour was usually of benefit. Thus, we were able to accept all hypotheses except H2 for T1. When a precise task is unknown, we suggest using orientation with colour encoding (ORI+COL).

The tasks where colour was most beneficial (T2, T3, T4) asked participants to judge the prevalence of a mark globally in the visualisation. This result agrees with similar results on node-link diagrams [4]. One possible explanation for the negative impact of

colour in T1 is that this task requires retrieving detailed information within a cell but marks do not need to be distinguished. In this case, the shape of the glyph (star-like shape in orientation) was the predominant factor while colour added unnecessary visual complexity. ORI might be scalable to more values through smaller adaptations in shape and position of bars [20]. POS is unlikely to be scalable. Future studies are required. Also, we would like to see if the results generalise to larger networks.

Recommendations—Based on our evaluations, we recommend the following to make the best use of our findings:

- R1: Orientation+Colour first**—use colour + orientation (ORI+COL).
- R2: Optional-colour**—if your interface is interactive, provide an option to turn colour on and off.
- R3: Avoid Opacity+Colour**—using opacity to encode edge weights in coloured glyphs can be problematic. We recommend using size (width or length) to encode edge weights when colour is also used in the glyph.
- R4: Connected-glyphs**—generally, choose glyphs and visual encodings that result in connected glyph designs. Fragmented glyphs become harder to perceive as one, especially when they are located next to other glyphs and occupy a small area. For example, in Figure 4 (c) and (d), the two glyphs between neighbouring cells (1,3) and (1,4) are hard to distinguish.
- R5: Grid**—for any encoding, leave margins between neighbouring matrix cells (i.e., glyphs) or use a salient visual grid.

Limitations and Future Work—Further experimentation is necessary. We have only tested four edge types. This decision was based on our domain which only required four Markov lags. BayesPiles [39] is very scalable in terms of the number of networks and in this paper we investigated how BayesPiles could be extended to dynamic Bayesian networks. Informed by this previous research and the specific requirements in the field, we focussed primarily on the visual encoding of individual networks. Our findings form a basis for future work that can deal with issues such as the scalability such as a larger number of edge types, edge weights, network density, and comparing more than two networks. Our proposed glyphs should also be evaluated for other tasks, such as identifying clusters or other patterns of interest.

9 CONCLUSION

In this research, we collaborated with computational biologists to create encodings for dynamic Bayesian networks. Our encodings apply to networks with multiple edge types. Our main contribution is a formal evaluation of encodings that can inform the design of visualisation tools for domains with multilayer networks. We ran four crowdsourced experiments to evaluate these encodings. The performance of our visual encodings was task-dependent. For more local tasks, we found that colour hindered performance, but for all other tasks, it improved performance. In all tasks, orientation outperformed position.

ACKNOWLEDGMENTS

We thank the Multilayer Nets Workshop at IEEE VIS 2019 for its valuable feedback.

REFERENCES

- [1] Andrej Aderhold, V. Anne Smith, and Dirk Husmeier. 2015. *Biological Network Inference at Multiple Scales: From Gene Regulation to Species Interactions*. John Wiley & Sons, Ltd, Chapter 27, 525–554. <https://doi.org/10.1002/9781119078845.ch27>
- [2] Basak Alper, Benjamin Bach, Nathalie Henry Riche, Tobias Isenberg, and Jean-Daniel Fekete. 2013. Weighted Graph Comparison Techniques for Brain Connectivity Analysis (*CHI '13*). 483–492. <https://doi.org/10.1145/2470654.2470724>
- [3] Daniel Archambault, Tamara Munzner, and David Auber. 2008. GrouseFlocks: Steerable exploration of graph hierarchy space. *IEEE Transactions on Visualization and Computer Graphics* 14, 4 (2008), 900–913. <https://doi.org/10.1109/tvcg.2008.34>
- [4] Daniel Archambault, Helen C Purchase, and Bruno Pinaud. 2010. The Readability of Path-Preserving Clusterings of Graphs. *Computer Graphics Forum* 29, 3 (2010), 1173–1182. <https://doi.org/10.1111/j.1467-8659.2009.01683.x>
- [5] Benjamin Bach, Nathalie Henry-Riche, Tim Dwyer, Tara Madhyastha, J-D Fekete, and Thomas Grabowski. 2015. Small MultiPiles: Piling Time to Explore Temporal Patterns in Dynamic Networks. *Computer Graphics Forum* 34, 3 (2015), 31–40. <https://doi.org/10.1111/cgf.12615>
- [6] Benjamin Bach, Emmanuel Pietriga, and Jean-Daniel Fekete. 2014. Visualizing Dynamic Networks with Matrix Cubes (*CHI '14*). ACM, New York, NY, USA, 877–886. <https://doi.org/10.1145/2556288.2557010>
- [7] Benjamin Bach, Emmanuel Pietriga, Ilaria Liccardi, and Gennady Legostaev. 2013. Visualizing populated ontologies with OntoTrix. *International Journal On Semantic Web and Information Systems (IJSWIS)* 9, 4 (2013), 17–40.
- [8] Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. 2014. The State of the Art in Visualizing Dynamic Graphs. In *EuroVis - STARS*, R. Borgo, R. Maciejewski, and I. Viola (Eds.). The Eurographics Association. <https://doi.org/10.2312/eurovisstar.20141174>
- [9] Fabian Beck and Stephan Diehl. 2013. Visual comparison of software architectures. *Information Visualization* 12, 2 (2013), 178–199. <https://doi.org/10.1177/1473871612455983>
- [10] Rita Borgo, Johannes Kehler, David HS Chung, Eamonn Maguire, Robert S Laramée, Helwig Hauser, Matthew Ward, and Min Chen. 2013. Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications. In *Eurographics 2013 - State of the Art Reports*, M. Sbert and L. Szirmay-Kalos (Eds.). The Eurographics Association. <https://doi.org/10.2312/conf/EG2013/stars/039-063>
- [11] Ulrik Brandes and Bobo Nick. 2011. Asymmetric Relations in Longitudinal Social Networks. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (Dec 2011), 2283–2290. <https://doi.org/10.1109/TVCG.2011.169>
- [12] Ulrik Brandes, Bobo Nick, Brigitte Rockstroh, and Astrid Steffen. 2013. Gestaltlines. *Computer Graphics Forum* 32, 3pt2 (2013), 171–180. <https://doi.org/10.1111/cgf.12104>
- [13] Ricardo Cava, Carla M Freitas, and Marco Winckler. 2017. ClusterVis: Visualizing Nodes Attributes in Multivariate Graphs (*SAC '17*). ACM, New York, NY, USA, 174–179. <https://doi.org/10.1145/3019612.3019684>
- [14] Chunlei Chang, Benjamin Bach, Tim Dwyer, and Kim Marriott. 2017. Evaluating Perceptually Complementary Views for Network Exploration Tasks (*CHI '17*). ACM, New York, NY, USA, 1397–1407. <https://doi.org/10.1145/3025453.3026024>
- [15] Tuan Nhon Dang, Paul Murray, and Angus Graeme Forbes. 2015. PathwayMatrix: Visualizing binary relationships between proteins in biological pathways. In *BMC proc.*, Vol. 9. BioMed Central, S3. <https://doi.org/10.1186/1755-6561-9-S6-S3>
- [16] Cody Dunne, Nathalie Henry Riche, Bongshin Lee, Ronald Metoyer, and George Robertson. 2012. GraphTrail: Analyzing Large Multivariate, Heterogeneous Networks While Supporting Exploration History (*CHI '12*). ACM, New York, NY, USA, 1663–1672. <https://doi.org/10.1145/2207676.2208293>
- [17] Cody Dunne and Ben Shneiderman. 2013. Motif Simplification: Improving Network Visualization Readability with Fan, Connector, and Clique Glyphs (*CHI '13*). ACM, New York, NY, USA, 3247–3256. <https://doi.org/10.1145/2470654.2466444>
- [18] Tim Dwyer, Nathalie Henry Riche, Kim Marriott, and Christopher Mears. 2013. Edge Compression Techniques for Visualization of Dense Directed Graphs. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (Dec 2013), 2596–2605. <https://doi.org/10.1109/TVCG.2013.151>
- [19] Niklas Elmqvist, Thanh-Nghi Do, Howard Goodell, Nathalie Henry, and Jean-Daniel Fekete. 2008. ZAME: Interactive large-scale graph visualization. In *2008 IEEE Pacific visualization symposium*. IEEE, 215–222.
- [20] Johannes Fuchs, Fabian Fischer, Florian Mansmann, Enrico Bertini, and Petra Isenberg. 2013. Evaluation of Alternative Glyph Designs for Time Series Data in a Small Multiple Setting (*CHI '13*). ACM, New York, NY, USA, 3237–3246. <https://doi.org/10.1145/2470654.2466443>
- [21] Johannes Fuchs, Petra Isenberg, Anastasia Bezerianos, and Daniel Keim. 2017. A Systematic Review of Experimental Studies on Data Glyphs. *IEEE Transactions on Visualization and Computer Graphics* 23, 7 (July 2017), 1863–1879. <https://doi.org/10.1109/TVCG.2016.2549018>
- [22] Mohammad Ghoniem, Jean-Daniel Fekete, and Philippe Castagliola. 2005. On the Readability of Graphs Using Node-Link and Matrix-Based Representations: A Controlled Experiment and Statistical Analysis. *Information Visualization* 4, 2 (2005), 114–135. <https://doi.org/10.1057/palgrave.ivs.9500092>
- [23] Michael Gleicher, Danielle Albers, Rick Walker, Ilir Jusufi, Charles D Hansen, and Jonathan C Roberts. 2011. Visual comparison for information visualization. *Information Visualization* 10, 4 (2011), 289–309. <https://doi.org/10.1177/1473871611416549>
- [24] Hua Guo, Jeff Huang, and David H Laidlaw. 2015. Representing Uncertainty in Graph Edges: An Evaluation of Paired Visual Variables. *IEEE Transactions on Visualization and Computer Graphics* 21, 10 (Oct 2015), 1173–1186. <https://doi.org/10.1109/TVCG.2015.2424872>
- [25] Aric Hagberg, Pieter Swart, and Daniel S Chult. 2008. *Exploring network structure, dynamics, and function using NetworkX*. Technical Report. Los Alamos National Lab.(LANL), Los Alamos, NM (United States).
- [26] Mark Harrower and Cynthia A. Brewer. 2011. *ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps*. John Wiley & Sons, Ltd, Chapter 3.8, 261–268. <https://doi.org/10.1002/9780470979587.ch34>
- [27] Christopher Kintzel, Johannes Fuchs, and Florian Mansmann. 2011. Monitoring Large IP Spaces with ClockView (*VizSec '11*). Association for Computing Machinery, New York, NY, USA, Article Article 2, 10 pages. <https://doi.org/10.1145/2016904.2016906>
- [28] Jock Mackinlay. 1986. Automating the Design of Graphical Presentations of Relational Information. *ACM Trans. Graph.* 5, 2 (April 1986), 110–141. <https://doi.org/10.1145/22949.22950>
- [29] Jon May and Philip J Barnard. 1997. Modelling Multimodal Interaction: A Theory-based Technique for Design Analysis Support. In *Proceedings of the IFIP TC13 International Conference on Human-Computer Interaction (INTERACT '97)*. Chapman & Hall, Ltd., London, UK, UK, 667–668.
- [30] Fintan McGehee, Mohammad Ghoniem, Guy Melançon, Benoit Otjacques, and Bruno Pinaud. 2019. The State of the Art in Multilayer Network Visualization. <https://doi.org/10.1111/cgf.13610>
- [31] Kevin Patrick Murphy. 2002. *Dynamic Bayesian Networks: Representation, Inference and Learning*. Ph.D. Dissertation. AAI3082340.
- [32] Mershack Okoe, Radu Jianu, and Stephen G Kobourov. 2019. Node-link or Adjacency Matrices: Old Question, New Insights. *IEEE Transactions on Visualization and Computer Graphics* (2019). <https://doi.org/10.1109/TVCG.2018.2865940>
- [33] A Johannes Pretorius and Jarke J Van Wijk. 2008. Visual Inspection of Multivariate Graphs. *Computer Graphics Forum* 27, 3 (2008), 967–974. <https://doi.org/10.1111/j.1467-8659.2008.01231.x>
- [34] Benjamin Renoust, Guy Melançon, and Tamara Munzner. 2015. Detangler: Visual Analytics for Multiplex Networks. *Computer Graphics Forum* 34, 3 (2015), 321–330. <https://doi.org/10.1111/cgf.12644>
- [35] Nathalie Henry Riche, Tim Dwyer, Bongshin Lee, and Sheelagh Carpendale. 2012. Exploring the Design Space of Interactive Link Curvature in Network Diagrams (*AVI '12*). ACM, New York, NY, USA, 506–513. <https://doi.org/10.1145/2254556.2254652>
- [36] V Anne Smith, Jing Yu, Tom V Smulders, Alexander J Hartemink, and Erich D Jarvis. 2006. Computational Inference of Neural Information Flow Networks. *PLOS Computational Biology* 2, 11 (11 2006), 1–14. <https://doi.org/10.1371/journal.pcbi.0020161>
- [37] Edward R. Tufte. 1986. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA.
- [38] Stef Van den Elzen and Jarke J Van Wijk. 2014. Multivariate Network Exploration and Presentation: From Detail to Overview via Selections and Aggregations. *IEEE Transactions on Visualization and Computer Graphics* 20, 12 (Dec 2014), 2310–2319. <https://doi.org/10.1109/TVCG.2014.2346441>
- [39] Athanasios Vogogias, Jessie Kennedy, Daniel Archambault, Benjamin Bach, V Anne Smith, and Hannah Currant. 2018. BayesPiles: Visualisation Support for Bayesian Network Structure Learning. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 1, Article 5 (Nov 2018), 23 pages. <https://doi.org/10.1145/3230623>
- [40] Tatiana Von Landesberger, Arjan Kuijper, Tobias Schreck, Jörn Kohlhammer, Jarke J van Wijk, J-D Fekete, and Dieter W Fellner. 2011. Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges. *Computer Graphics Forum* 30, 6 (2011), 1719–1749. <https://doi.org/10.1111/j.1467-8659.2011.01898.x>
- [41] Matthew O. Ward. 2008. *Multivariate Data Glyphs: Principles and Practice*. Springer Berlin Heidelberg, Berlin, Heidelberg, 179–198. https://doi.org/10.1007/978-3-540-33037-0_8
- [42] Martin Wattenberg. 2006. Visual Exploration of Multivariate Graphs (*CHI '06*). ACM, New York, NY, USA, 811–819. <https://doi.org/10.1145/1124772.1124891>
- [43] Zuoshuang Xiang, Rebecca M Minter, Xiaoming Bi, Peter J Woolf, and Yongqun He. 2007. miniTUBA: medical inference by network integration of temporal data using Bayesian analysis. *Bioinformatics* 23, 18 (07 2007), 2423–2432. <https://doi.org/10.1093/bioinformatics/btm372>
- [44] Vahan Yeghordjian, Daniel Archambault, Stephan Diehl, Tim Dwyer, Karsten Klein, Helen C. Purchase, and Hsiang-Yun Wu. 2018. Exploring the limits of complexity: A survey of empirical studies on graph visualisation. *Visual Informatics* 2, 4 (2018), 264 – 282. <https://doi.org/10.1016/j.visinf.2018.12.006>
- [45] Jing Yu, V Anne Smith, Paul P Wang, Alexander J Hartemink, and Erich D Jarvis. 2002. Using Bayesian network inference algorithms to recover molecular genetic regulatory networks. In *International Conference on Systems Biology*, Vol. 2002.